임베디드 Week7 HW

2020161084 이도현

p.10-1

import RPi.GPIO as GPIO

import time

SW1 = 5

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BCM)

GPIO.setup(SW1, GPIO.IN, pull\_up\_down=GPIO.PUD\_DOWN)

try:

    while True:

        sw1Value = GPIO.input(SW1)

        if sw1Value == 1:

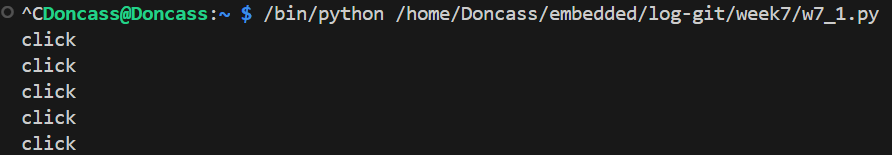
            print("click")

        time.sleep(0.1)

except KeyboardInterrupt:

    pass

GPIO.cleanup()



총 3번 눌렀는데 1번에 2번 click이 출력되는 일도 일어나서 5번의 click 출력 발생

p.10-2

import RPi.GPIO as GPIO

import time

SW1 = 5

click\_count = 0

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BCM)

GPIO.setup(SW1, GPIO.IN, pull\_up\_down=GPIO.PUD\_DOWN)

try:

    while True:

        sw1Value = GPIO.input(SW1)

        if sw1Value == 1:

            click\_count += 1

            print(f"click {click\_count}")

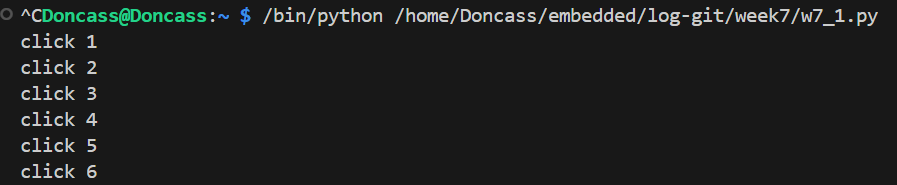
            time.sleep(0.1)

        time.sleep(0.1)

except KeyboardInterrupt:

    pass

GPIO.cleanup()



마찬가지로 3번 눌렀는데 6번의 click 출력

p.10-3

import RPi.GPIO as GPIO

import time

SW1 = 5

click\_count = 0

prev\_sw1Value = 0

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BCM)

GPIO.setup(SW1, GPIO.IN, pull\_up\_down=GPIO.PUD\_DOWN)

try:

    while True:

        sw1Value = GPIO.input(SW1)

        if prev\_sw1Value == 0 and sw1Value == 1:

            click\_count += 1

            print(f"click {click\_count}")

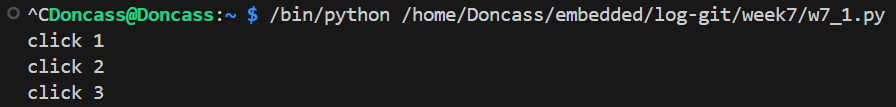
        prev\_sw1Value = sw1Value

        time.sleep(0.1)

except KeyboardInterrupt:

    pass

GPIO.cleanup()



눌렀을 때만 클릭 출력되게 변경하여 3회의 클릭에 3번의 click 출력 발생

p.10-4

import RPi.GPIO as GPIO

import time

# 스위치와 GPIO 핀 매핑

SW1\_PIN = 5    # GPIO05

SW2\_PIN = 6    # GPIO06

SW3\_PIN = 13   # GPIO13

SW4\_PIN = 19   # GPIO19

SW\_PINS = [SW1\_PIN, SW2\_PIN, SW3\_PIN, SW4\_PIN]

click\_counts = [0, 0, 0, 0]

prev\_values = [0, 0, 0, 0]

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BCM)

for pin in SW\_PINS:

    GPIO.setup(pin, GPIO.IN, pull\_up\_down=GPIO.PUD\_DOWN)

try:

    while True:

        for idx, pin in enumerate(SW\_PINS):

            value = GPIO.input(pin)

            if prev\_values[idx] == 0 and value == 1:

                click\_counts[idx] += 1

                print(f"('SW{idx+1} click', {click\_counts[idx]})")

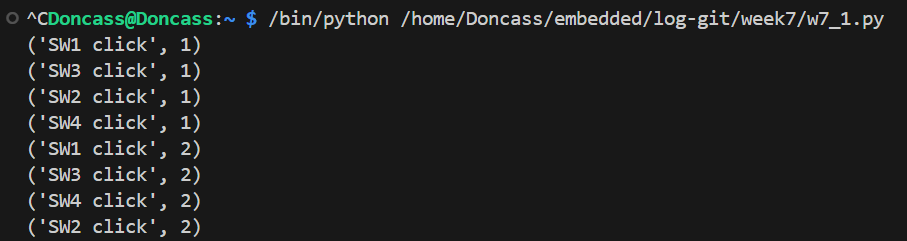
            prev\_values[idx] = value

        time.sleep(0.1)

except KeyboardInterrupt:

    pass

GPIO.cleanup()



p.14

import RPi.GPIO as GPIO

import time

BUZZER = 12

SWITCH = 5

# 각 음계의 주파수 (Hz)

scale = [261, 293, 329, 349, 392, 440, 493, 523]  # 도, 레, 미, 파, 솔, 라, 시, 도

horn = [

    (523, 0.2),  # 도

    (0, 0.1),    # 휴식 (무음)

    (523, 0.2),  # 도

    (0, 0.1),

    (659, 0.5),  # 미

]

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BCM)

GPIO.setup(BUZZER, GPIO.OUT)

GPIO.setup(SWITCH, GPIO.IN, pull\_up\_down=GPIO.PUD\_DOWN)

p = GPIO.PWM(BUZZER, 100)  # 초기 주파수는 임의로 설정

p.start(50)

# # 1. "도레미파솔라시도" 음계를 출력

# try:

#     for freq in scale:

#         p.ChangeFrequency(freq)

#         time.sleep(0.5)  # 각 음을 0.5초씩 재생

#     p.stop()

# except KeyboardInterrupt:

#     pass

# p.stop()

# GPIO.cleanup()

# # 2. 나만의 경적 소리 구현

# try:

#     for freq, duration in horn:

#         if freq == 0:

#             p.ChangeDutyCycle(0)  # 무음 처리

#         else:

#             p.ChangeFrequency(freq)

#             p.ChangeDutyCycle(50)

#         time.sleep(duration)

#     p.stop()

# except KeyboardInterrupt:

#     pass

# p.stop()

# GPIO.cleanup()

# 3. 스위치를 한 번 누르면 경적 소리가 나도록 구현

# p.stop()

# prev\_input = 0

# try:

#     while True:

#         input = GPIO.input(SWITCH)

#         if prev\_input == 0 and input == 1:

#             # 스위치가 눌렸을 때 경적 소리 재생

#             p.start(50)

#             for freq, duration in horn:

#                 if freq == 0:

#                     p.ChangeDutyCycle(0)

#                 else:

#                     p.ChangeFrequency(freq)

#                     p.ChangeDutyCycle(50)

#                 time.sleep(duration)

#             p.stop()

#         prev\_input = input

#         time.sleep(0.05)

# except KeyboardInterrupt:

#     pass

# p.stop()

# GPIO.cleanup()

# 4. 스위치 4개를 사용하여 나만의 음악을 연주

SW\_PINS = [5, 6, 13, 19]

notes = [261, 293, 329, 349]  # 도, 레, 미, 파

for pin in SW\_PINS:

    GPIO.setup(pin, GPIO.IN, pull\_up\_down=GPIO.PUD\_DOWN)

p.stop()  # 초기에는 소리를 내지 않음

prev\_inputs = [0, 0, 0, 0]

try:

    while True:

        for idx, pin in enumerate(SW\_PINS):

            input = GPIO.input(pin)

            if prev\_inputs[idx] == 0 and input == 1:

                # 스위치가 눌렸을 때 해당 음계 재생

                freq = notes[idx]

                p.start(50)

                p.ChangeFrequency(freq)

                time.sleep(0.5)

                p.stop()

            prev\_inputs[idx] = input

        time.sleep(0.05)

except KeyboardInterrupt:

    pass

p.stop()

GPIO.cleanup()

p.17-1

import RPi.GPIO as GPIO

import time

# 왼쪽 모터 핀 정의

PWMA = 18

AIN1 = 22

AIN2 = 27

# 오른쪽 모터 핀 정의

PWMB = 23

AIN3 = 24

AIN4 = 25

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BCM)

# 왼쪽 모터 설정

GPIO.setup(PWMA, GPIO.OUT)

GPIO.setup(AIN1, GPIO.OUT)

GPIO.setup(AIN2, GPIO.OUT)

# 오른쪽 모터 설정

GPIO.setup(PWMB, GPIO.OUT)

GPIO.setup(AIN3, GPIO.OUT)

GPIO.setup(AIN4, GPIO.OUT)

# 왼쪽 및 오른쪽 모터의 PWM 초기화

L\_Motor = GPIO.PWM(PWMA, 500)

R\_Motor = GPIO.PWM(PWMB, 500)

L\_Motor.start(0)

R\_Motor.start(0)

try:

    while True:

        # 왼쪽 모터: 정방향으로 최대 속도(100%)

        GPIO.output(AIN1, 0)

        GPIO.output(AIN2, 1)

        L\_Motor.ChangeDutyCycle(100)

        # 오른쪽 모터: 정방향으로 50% 속도

        GPIO.output(AIN3, 1)

        GPIO.output(AIN4, 0)

        R\_Motor.ChangeDutyCycle(50)

        time.sleep(1.0)

        # 두 모터 모두 정지

        L\_Motor.ChangeDutyCycle(0)

        R\_Motor.ChangeDutyCycle(0)

        time.sleep(1.0)

except KeyboardInterrupt:

    pass

GPIO.cleanup()

p.17-2

import RPi.GPIO as GPIO

import time

# 모터 핀 설정

PWMA = 18  # 왼쪽 모터

AIN1 = 22

AIN2 = 27

PWMB = 23  # 오른쪽 모터

AIN3 = 24

AIN4 = 25

# 스위치 핀 설정

SW1 = 5    # 앞

SW2 = 6    # 오른쪽

SW3 = 13   # 왼쪽

SW4 = 19   # 뒤

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BCM)

# 모터 핀 설정

GPIO.setup(PWMA, GPIO.OUT)

GPIO.setup(AIN1, GPIO.OUT)

GPIO.setup(AIN2, GPIO.OUT)

GPIO.setup(PWMB, GPIO.OUT)

GPIO.setup(AIN3, GPIO.OUT)

GPIO.setup(AIN4, GPIO.OUT)

# 스위치 핀 설정

GPIO.setup(SW1, GPIO.IN, pull\_up\_down=GPIO.PUD\_DOWN)

GPIO.setup(SW2, GPIO.IN, pull\_up\_down=GPIO.PUD\_DOWN)

GPIO.setup(SW3, GPIO.IN, pull\_up\_down=GPIO.PUD\_DOWN)

GPIO.setup(SW4, GPIO.IN, pull\_up\_down=GPIO.PUD\_DOWN)

# 모터 PWM 초기화

L\_Motor = GPIO.PWM(PWMA, 500)

R\_Motor = GPIO.PWM(PWMB, 500)

L\_Motor.start(0)

R\_Motor.start(0)

try:

    while True:

        if GPIO.input(SW1) == GPIO.HIGH:  # 앞쪽 이동

            print("SW1 눌림: 앞으로 이동")

            GPIO.output(AIN1, 0)

            GPIO.output(AIN2, 1)

            GPIO.output(AIN3, 1)

            GPIO.output(AIN4, 0)

            L\_Motor.ChangeDutyCycle(100)

            R\_Motor.ChangeDutyCycle(100)

        elif GPIO.input(SW2) == GPIO.HIGH:  # 오른쪽 이동

            print("SW2 눌림: 오른쪽으로 이동")

            GPIO.output(AIN1, 0)

            GPIO.output(AIN2, 1)

            GPIO.output(AIN3, 0)

            GPIO.output(AIN4, 1)

            L\_Motor.ChangeDutyCycle(50)

            R\_Motor.ChangeDutyCycle(50)

        elif GPIO.input(SW3) == GPIO.HIGH:  # 왼쪽 이동

            print("SW3 눌림: 왼쪽으로 이동")

            GPIO.output(AIN1, 1)

            GPIO.output(AIN2, 0)

            GPIO.output(AIN3, 1)

            GPIO.output(AIN4, 0)

            L\_Motor.ChangeDutyCycle(50)

            R\_Motor.ChangeDutyCycle(50)

        elif GPIO.input(SW4) == GPIO.HIGH:  # 뒤로 이동

            print("SW4 눌림: 뒤로 이동")

            GPIO.output(AIN1, 1)

            GPIO.output(AIN2, 0)

            GPIO.output(AIN3, 0)

            GPIO.output(AIN4, 1)

            L\_Motor.ChangeDutyCycle(100)

            R\_Motor.ChangeDutyCycle(100)

        else:  # 스위치가 눌리지 않았을 때 정지

            L\_Motor.ChangeDutyCycle(0)

            R\_Motor.ChangeDutyCycle(0)

        time.sleep(0.1)  # 짧은 지연 시간

except KeyboardInterrupt:

    pass

GPIO.cleanup()

